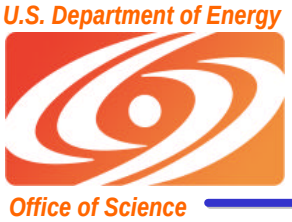


Secure Distributed Collaboration Capabilities and Infrastructure

Deb Agarwal
Distributed Systems Department

Lawrence Berkeley National Laboratory

Outline

- Background
- Collaborative Interaction Tools
  - Presence
  - Messaging
  - File sharing
- Transport mechanisms
  - High performance TCP
  - Group communication
- Security
  - Authentication
  - Authorization
  - Key agreement
  - Incremental Trust
- Grid middleware
- End-2-End Monitoring

U.S. Department of Energy

Office of Science

Scientific Collaboration Environment

BERKELEY LAB

- **Collaborative communication options**
  - **Formal meeting in person**
  - **Videoconference**
  - **Teleconference/telephone**
  - **Informal discussion/meeting**
  - **File/document sharing**
  - **E-mail/chat**
  - **Papers/documents/web**

Increasing % of time

Decreasing synchrony

Collaboration Realities

- Collaboration takes effort
  - Must provide a perceptible benefit
  - Must fit with current work practices
- Collaboration tools need to be used regularly (not on the shelf)
- Group must already have a strong need to collaborate

Collaborative Design Process

- Identify key activities to share
- Make sure all participants have an incentive
- Develop realistic use cases/interactions
- Role play the interactions
- Attempt the interaction using simple tools like the web or VNC and the telephone
- Identify critical missing elements
- Keep it as simple as possible
- Get in the habit of using it
- Support bootstrap efforts

U.S. Department of Energy

Office of Science

Pervasive Collaborative Computing Environment Goals

BERKELEY LAB

- Support ‘continuous’ collaboration
  - Ubiquitous – available anywhere
  - Synchronous and asynchronous
  - Persistent
- Low threshold for entry into the environment
- Target daily tasks and base connectivity
- Leverage off of existing components
- Secure environment
- Scale to support small and large groups

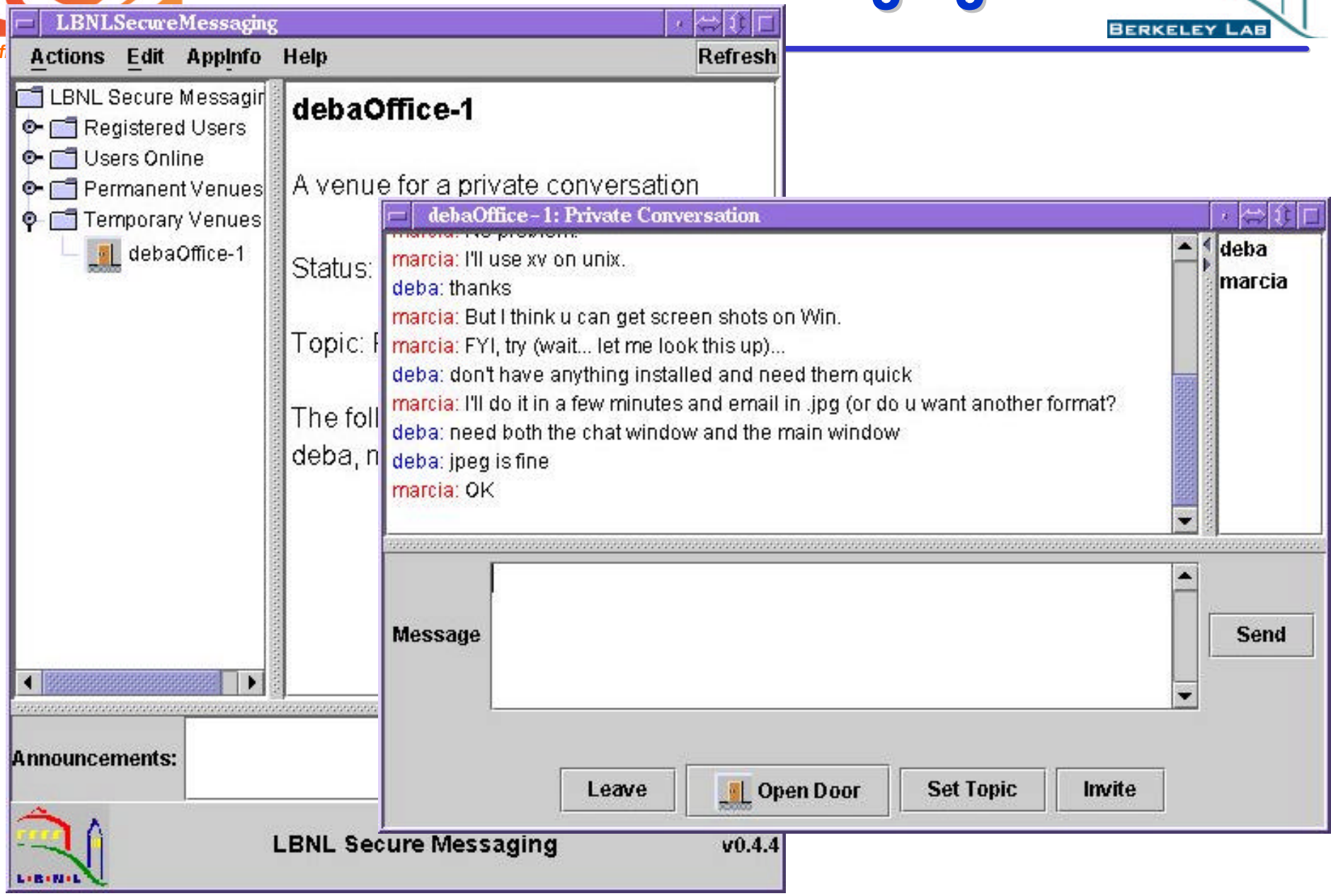
PCCE – Messaging

- Baseline presence information (rendezvous)
- Messaging
  - Permanent contexts topical meeting places
  - Group and private text-based messaging
  - Storage of preferences and current state
- Security
  - X.509 or username/password authentication
  - Data encryption using SSL connections
- Asynchronous notes

U.S. Department of Energy

PCCE - Secure Messaging

BERKELEY LAB



Peer-to-Peer File-Sharing

- Files shared from current locations
- Not dependent on servers
- Scalable to large number of users
- Distributed authorization
- Secure environment

Future Directions

- Collaborative workflow definition and tracking
- Shared editing
  - Code development
  - Text document
- Improved asynchronous messaging
- Shared applications
- Integration of videoconferencing

U.S. Department of Energy

Office of Science

Communication Protocols to Support Collaboratories

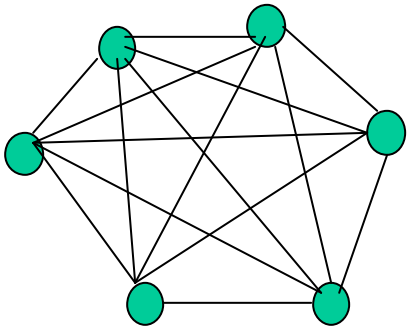
BERKELEY LAB

- Scalability
  - High data throughput
  - Large numbers of users interacting
- Support for peer-to-peer communication
- Robust – not dependent on servers

Group Communication

Scalability and robustness

- Provide efficient, reliable, and secure communication between collaborating sites
- Multicast communication channel directly connecting the participants
- Support participants spread across the Internet
- Support ad hoc formation of groups
- Remove dependence on servers



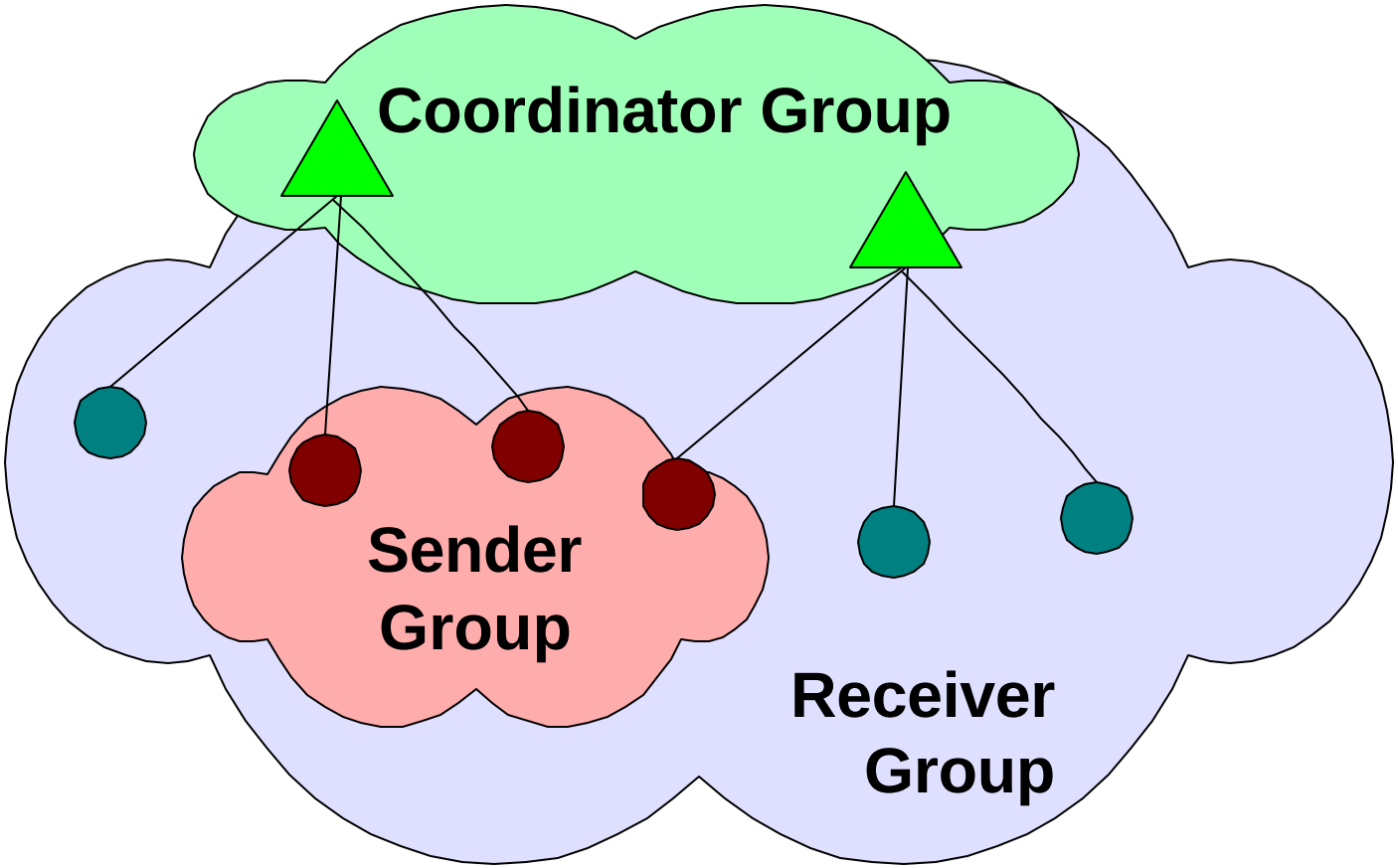
InterGroup Protocols

- **Goals**
  - **Support a broad range of applications**
    - Broadcast – one-to-many
    - Many-to-many
  - **Provide a broad range of guarantees**
    - Reliable and unreliable delivery
    - Sender order, total order, and unordered
  - **Based on IP Multicast**
- **Scale to the Internet**
  - **Many groups**
  - **Many members in each group**
  - **Heterogeneous latency between members**

The InterGroup Protocol

- Based on experience building past protocols
- Split group into a sender and receiver group
- Sender group membership
  - processes are in the sender group only while transmitting messages
  - strictly maintained
  - very dynamic
- Receiver group membership
  - not strictly maintained
  - hierarchically organized to scale to large groups
  - used for retransmissions and garbage collection

InterGroup Schematic



Improving TCP Performance

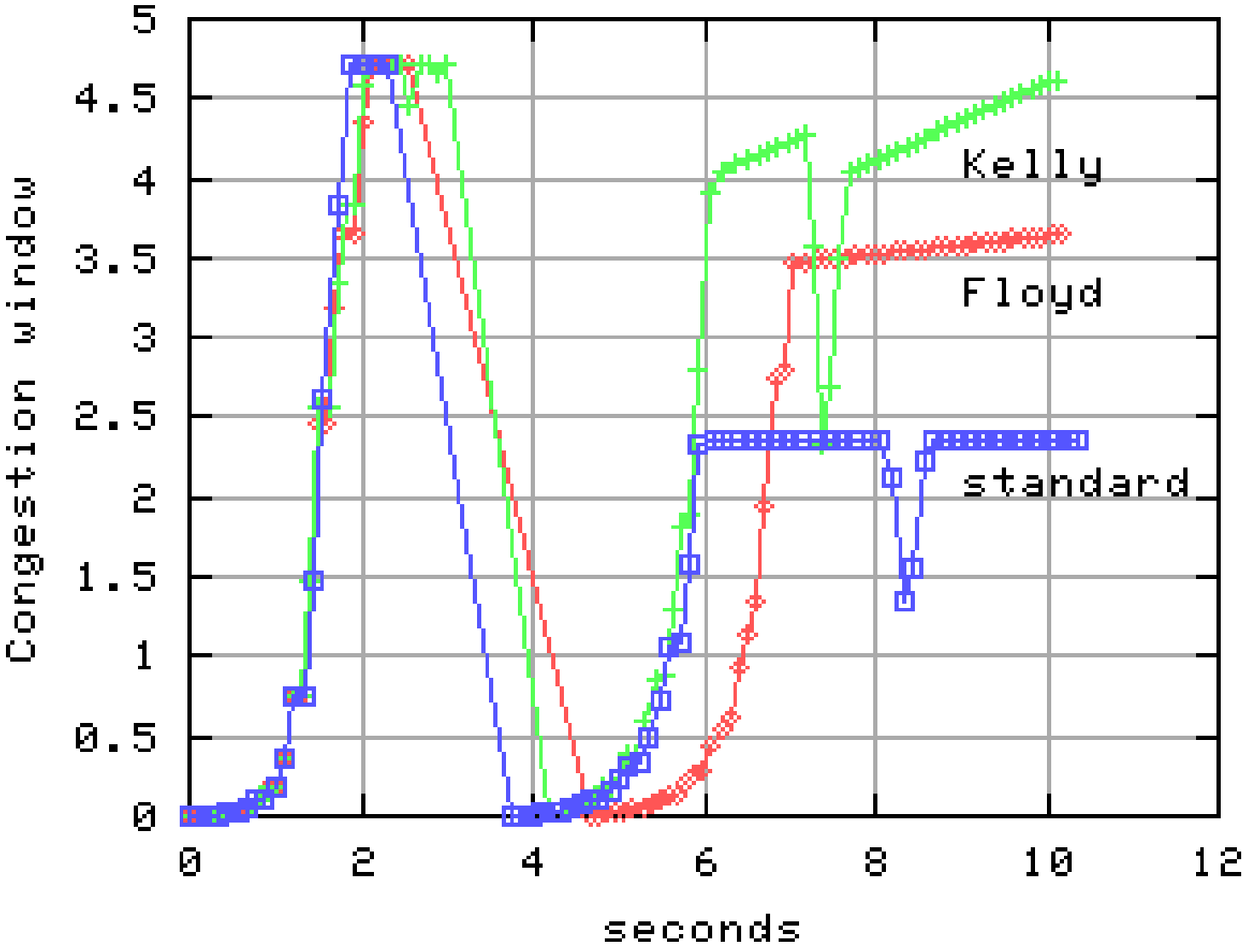
- Goal - improve performance on high bandwidth*delay product links
- HighSpeed TCP (proposed by Sally Floyd)
- Study based on simulations of the protocol
  - Bulk transfer capabilities
  - Fairness
  - Response to active queue management
- Results are very promising

U.S. Department of Energy

Office of Science

HighSpeed TCP Performance

BERKELEY LAB



Security

- Goals
  - Identify users – authentication
  - Define and enforce access control – authorization
  - Protect confidentiality of data – encryption
  - Define roles and levels of trust
  - Easy to configure and use from any location
- Tools
  - Akenti authorization server
  - Secure group layer
  - Message level security
  - Incremental trust

Akenti Goals

- Access based on policy statements made by multiple independent stakeholders
- Use Public Key Infrastructure (X.509) standards
  - To identify users
  - Create digitally signed certificates
  - Use TLS/GSI authenticated connections
- Emphasize usability
- Targeted at distributed environments
  - Users, resources, stakeholders are geographically and administratively distributed

Akenti Policy

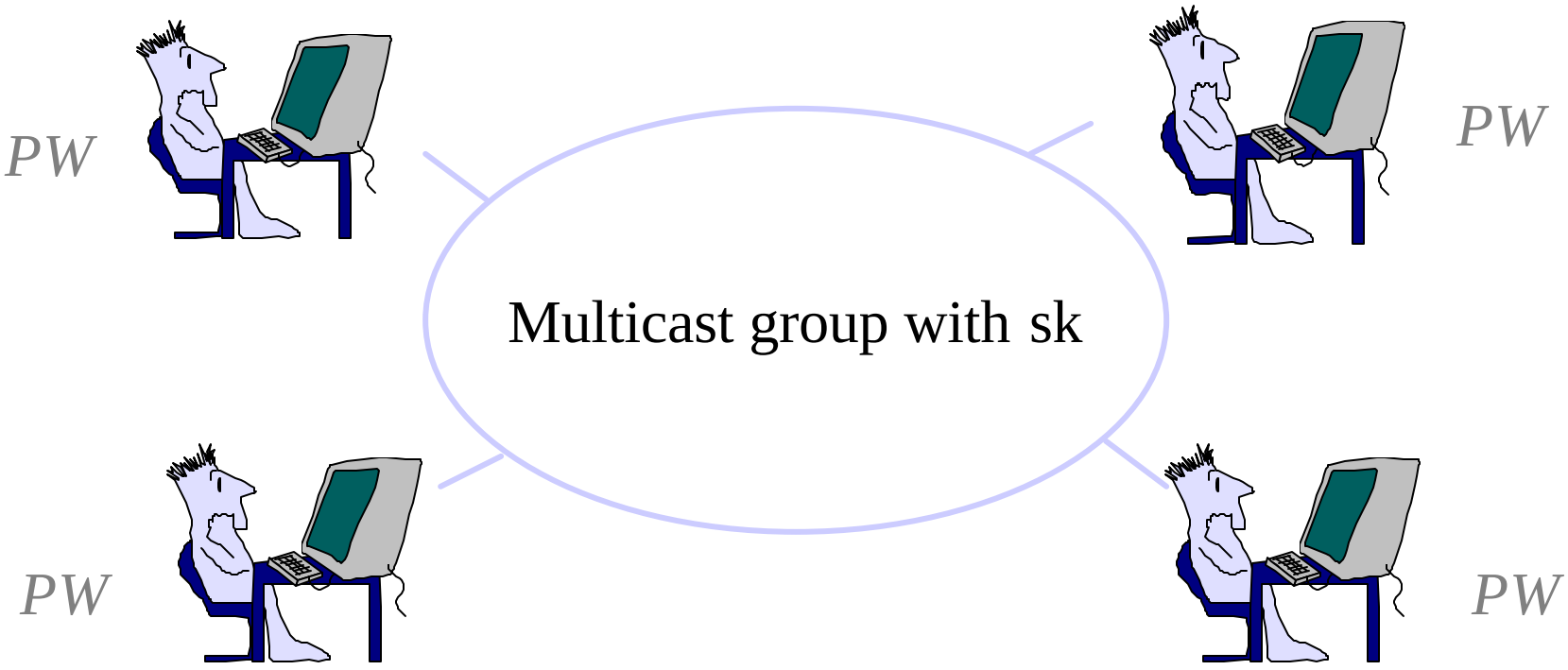
- Minimal local authorization policy files:
  - Who to trust, where to look for certificates.
- Most access control policy contained in distributed digitally signed certificates:
  - X.509 certificates for user identity and authentication
  - UseCondition certificates containing stakeholder policy
  - Attribute certificates in which a trusted party attests that a user possesses some attribute, e.g. training, group membership

Secure Group Layer - SGL

- Goals
  - Provide a secure channel for the group with properties similar to the SSL
  - Group authorization and access control is individually enforced
  - Fully distributed group key management (not centralized)
  - Portable implementation

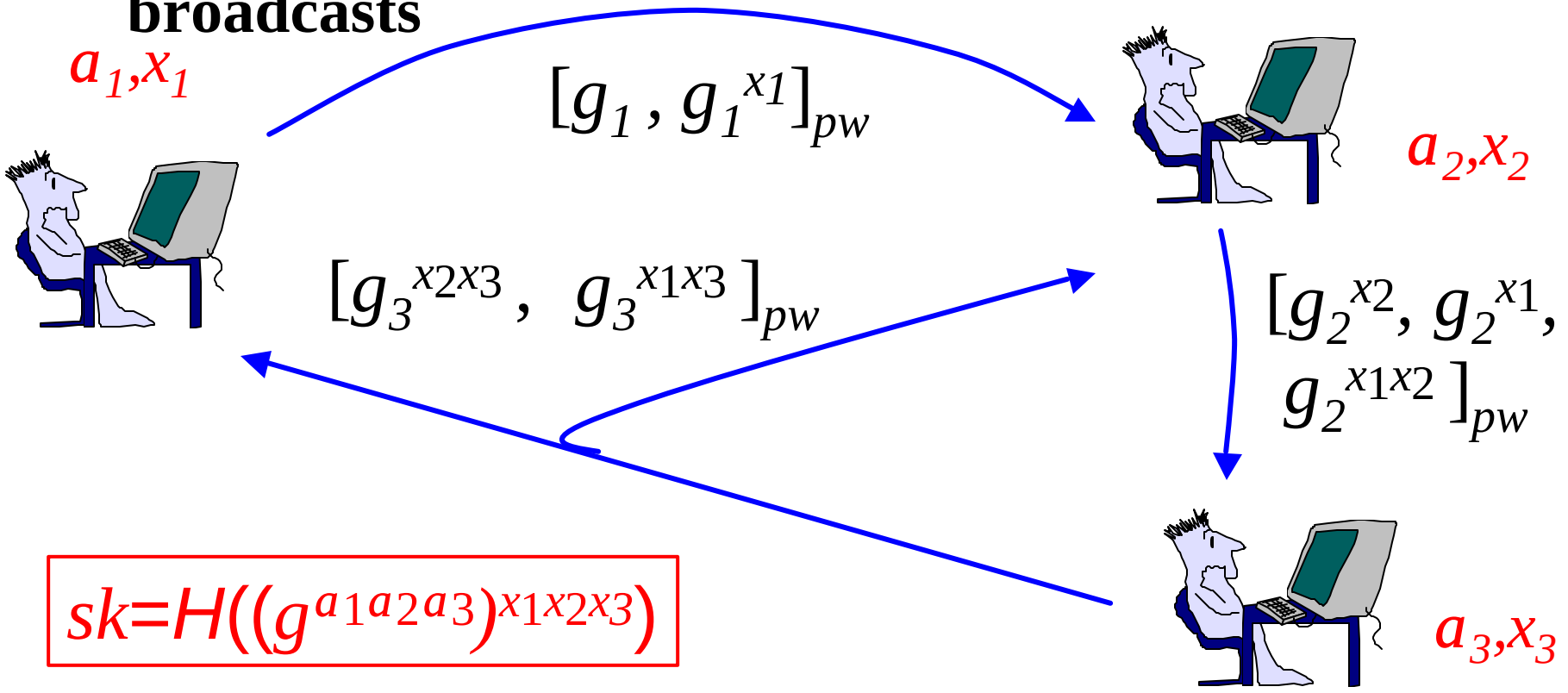
SGL Model of Communication

- **A set of *n* players**
  - **each player is represented by an oracle**
  - **each player holds a low-entropy secret (PW)**
- **A multicast group consisting of a set of players**



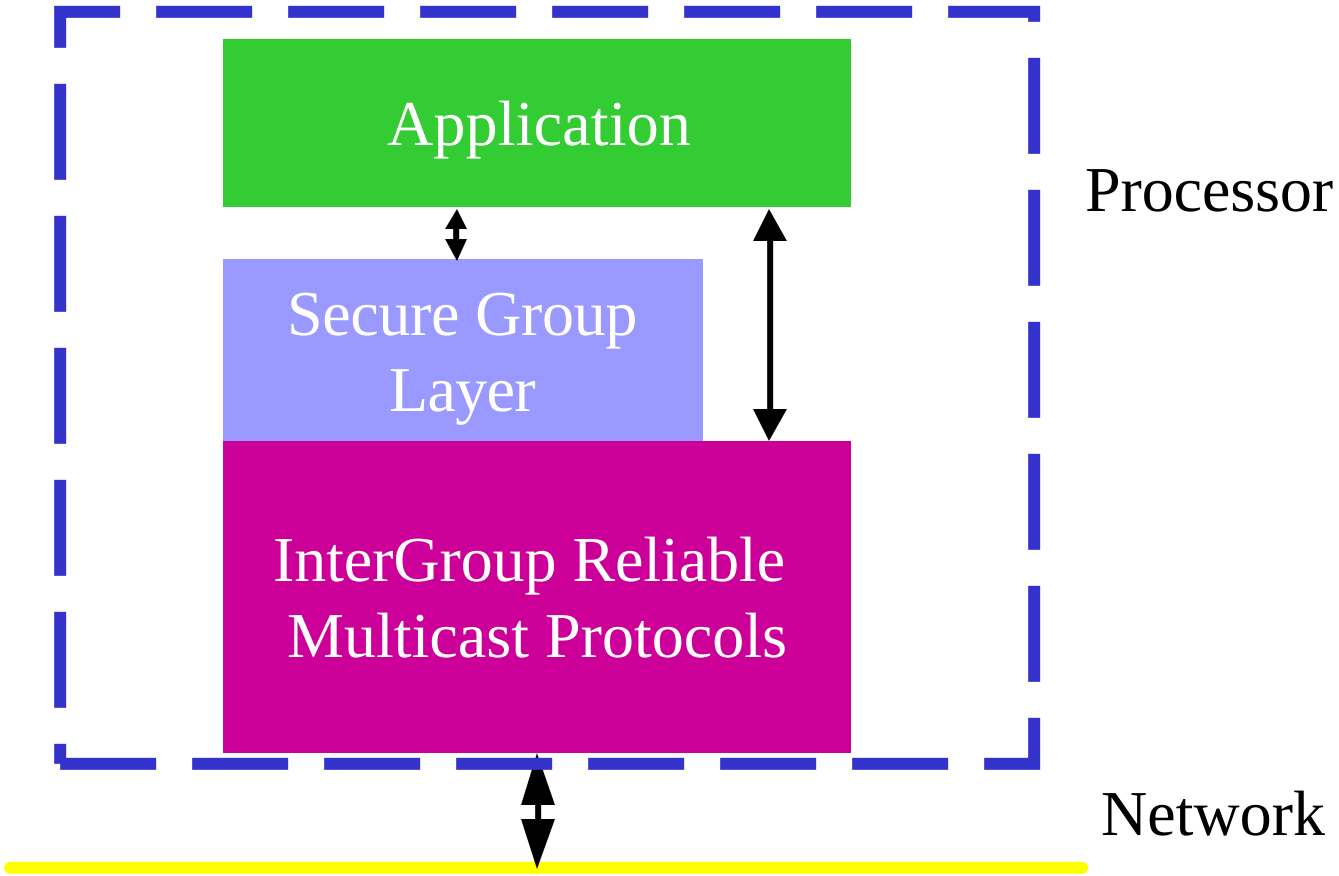
Group Diffie-Hellman Algorithm

- **Up-flow: U_i raises received values to the power of the values (x_i, a_i) and forwards to U_{i+1}**
- **Down-flow: U_n processes the last up-flow and broadcasts**



$$sk = H((g^{a_1 a_2 a_3})^{x_1 x_2 x_3})$$

InterGroup + SGL



Message Level Security

- Provide a *reliable* communication between an initial requestor and a Grid Services provider
  - message-level communication channel connecting the two entities
  - messages may be operated on by multiple intermediaries that perform actions (e.g, routing)

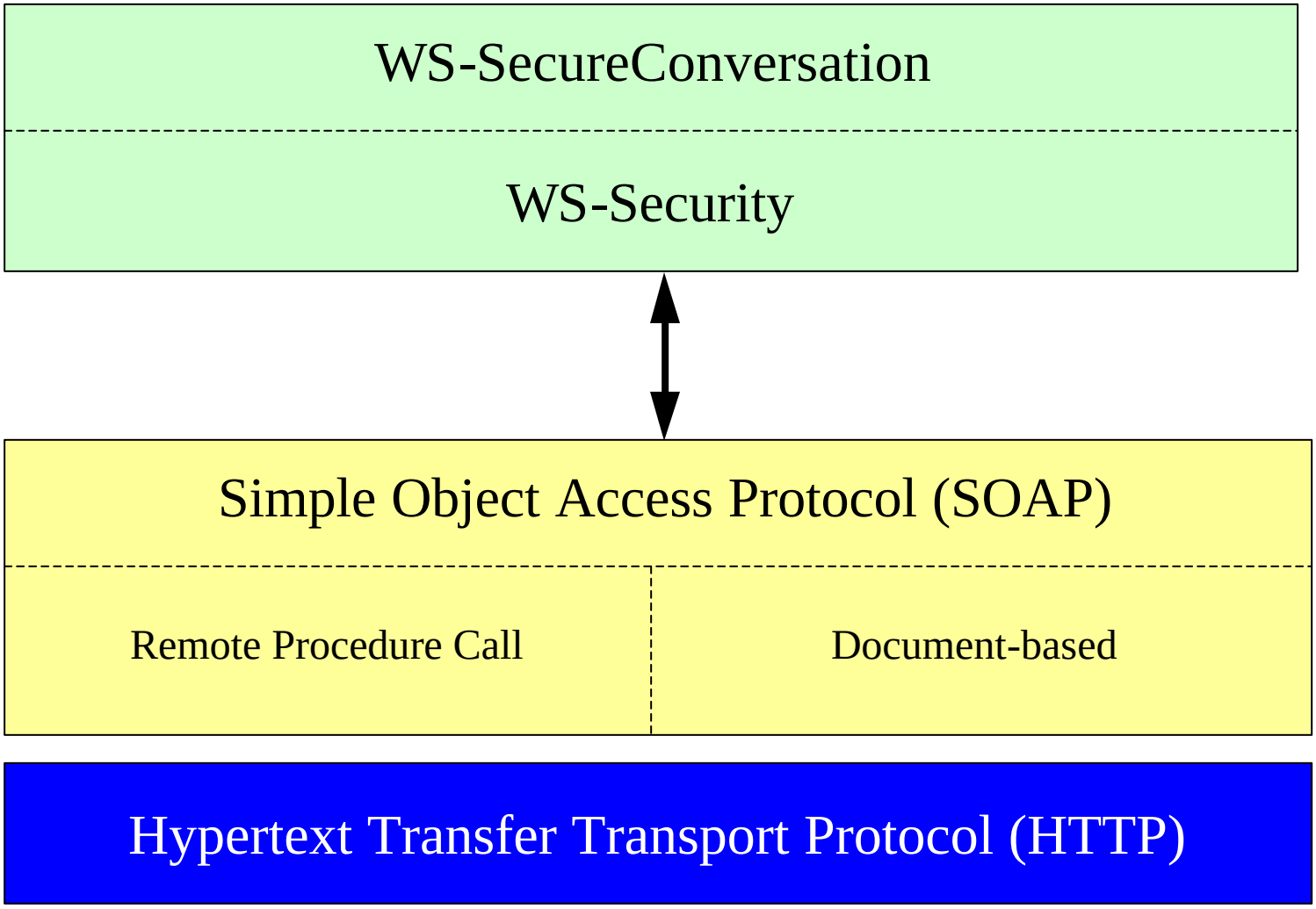
- Provide a *secure* communication between an initial requestor and a Grid Services provider
  - support confidentiality, authenticity, and integrity
  - support authorization and access control
  - support secure modification of messages operated on by intermediaries

U.S. Department of Energy

Office of Science

Security at the Application Layer : Architecture

BERKELEY LAB



U.S. Department of Energy
Office of Science

Security at the Application Layer: WS-SecureConversation

BERKELEY LAB

- The WS-SecureConversation component is similar to the TLS protocol but at the next level up (message-level)
  - confidentiality, authenticity, and integrity
  - authorization and access control
  - security services optional
- The WS-SecureConversation component establishes a *security* context between a requestor and a services provider
  - achieves confidentiality, authenticity, integrity using the WS-Security component
  - achieves authorization and access control using the WS-Policy and WS-Trust components
  - ➢ permits routing of SOAP messages without compromising the message security

Incremental Trust

- Allow new users to integrate into the collaboration gradually
- Support for levels of trust
- Multiple authentication methods

U.S. Department of Energy

Office of Science

Access to Grid Computing Capabilities

BERKELEY LAB

- On-demand simulations
- Distributed analyses
- Shared visualizations

Python CoG Kit

- Provide a mapping between Python and the Globus Toolkit®.
- Extend the use of Globus by enabling to access advanced Python features such as exceptions and objects for Grid programming.
- Grid portals

Grid Services Project

- **Develop Open Grid Services Architecture**
  - **Grid Services Specifications**
  - **Open Grid Services Infrastructure**
    - Implementations in multiple languages and protocols
  - **Higher-level Services**
  - **Application services and infrastructure**

Keeping an Eye on the System

- Monitor system performance
- Detect and identify problems
- Debug problems

End-2-End Monitoring The Problem

- **When building distributed systems, we often observe unexpectedly low performance**
  - the reasons for which are usually not obvious
- **The problems can be in any of the following components:**
  - the applications
  - the operating systems
  - the disks, network adapters, bus, memory, etc. on either the sending or receiving host
  - the network switches and routers, and so on

End-2-End Monitoring Goals

- **Improve end-to-end data throughput for data intensive applications**
- **Provide the ability to do performance analysis and fault detection**
- **Provide accurate, detailed, and adaptive monitoring of all of distributed computing components, including the network**

- **This requires a unified view of a wide range of sensor data, from network to host to application**
  - **define common protocols and data formats**
    - GGF efforts in this area
  - **work with others to integrate existing components using this framework**
    - e.g.: PingER, NWS, MDS
  - **develop missing pieces**
    - e.g.: event archives

End-2-End Example

- You know your end-to-end network path is all 622 Mbps or higher, but your file transfers are slow; what do you do?

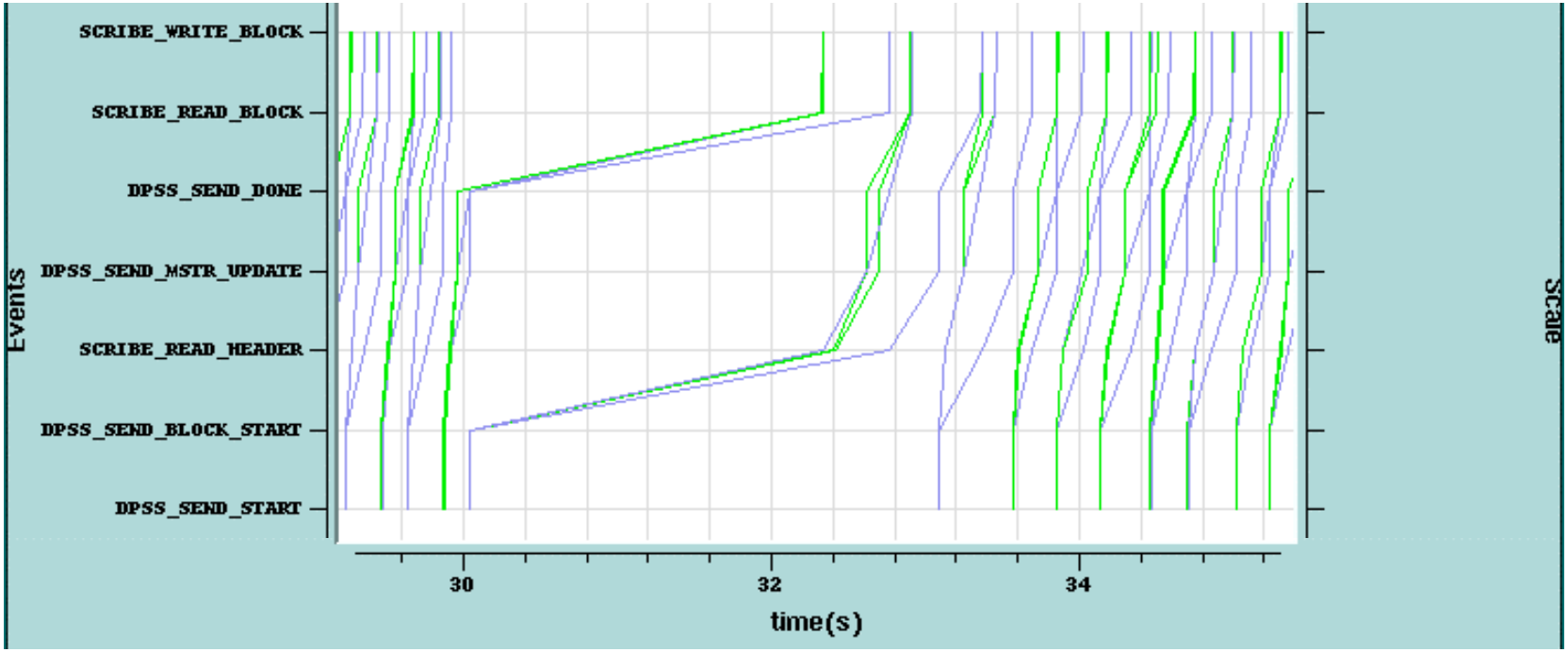
- Step 1: use “-bw” option to get performance results
  - E.g.: Tells you you are getting 10 Mbps

U.S. Department of Energy

Office of Science

Step 2: Add detailed application instrumentation

BERKELEY LAB

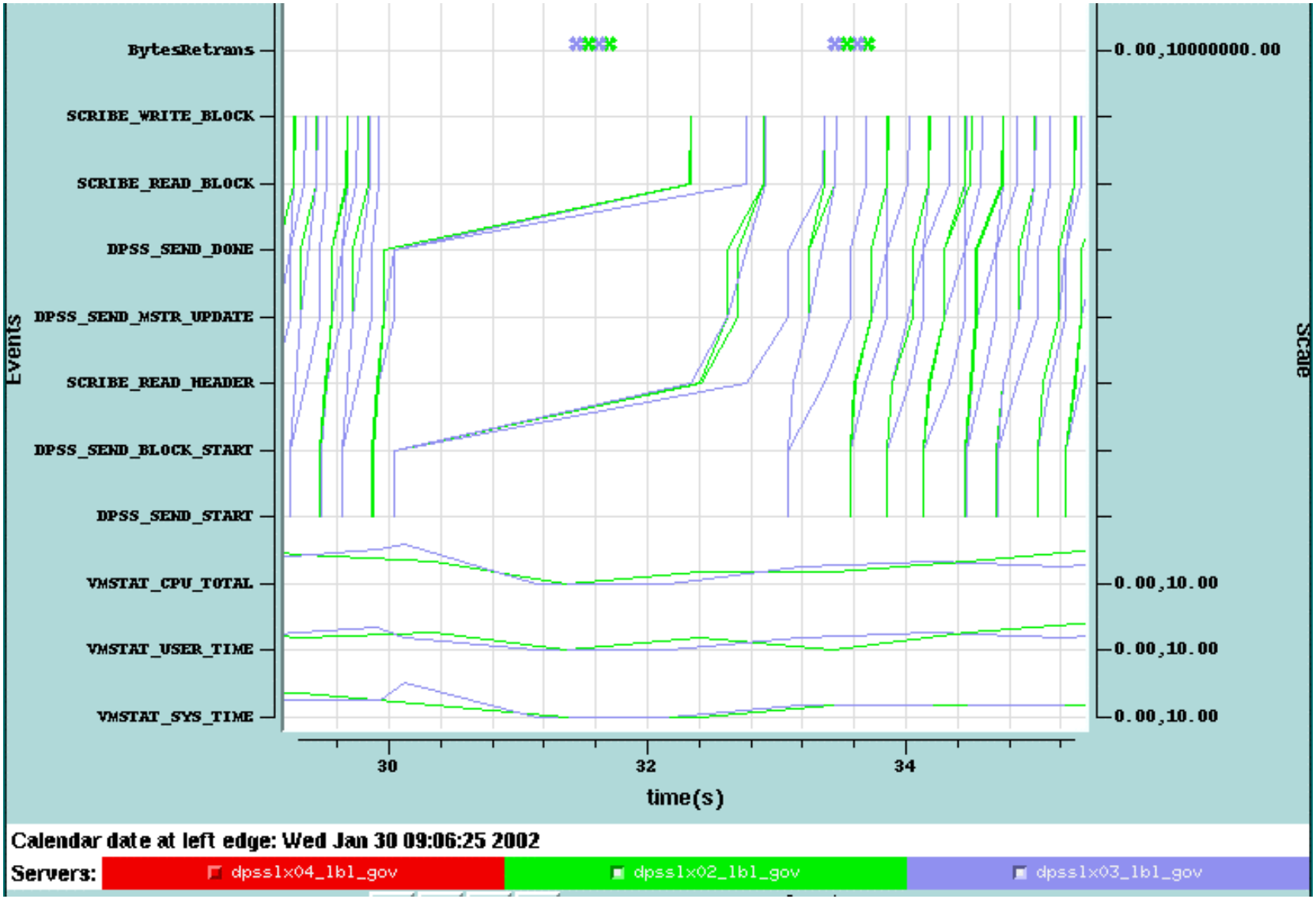


U.S. Department of Energy

Office of Science

Step 3: add host monitoring
(e.g.: CPU load and TCP retransmits)

BERKELEY LAB

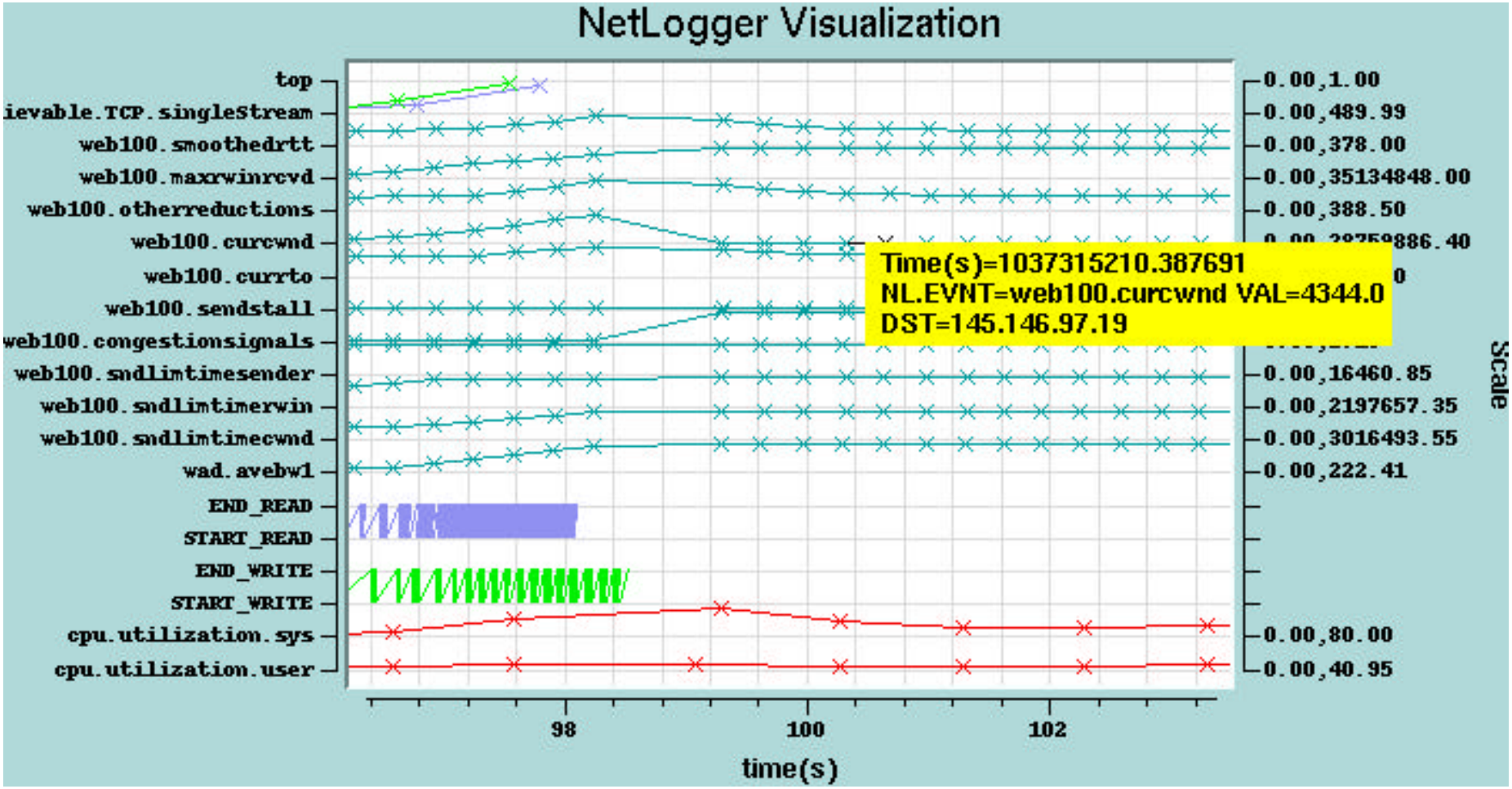


U.S. Department of Energy

Office of Science

Step 4: Add TCP instrumentation

BERKELEY LAB



Linux TCP SACK Bug: This happens when CWND is large and get manySACKs at one time

U.S. Department of Energy

Office of Science

Self Configuring Network Monitor (SCNM)

BERKELEY LAB

- **SCNM is a passive monitoring system designed to address the following issues:**
  - **Ability for network *users* to monitor their own traffic**
  - **Ability to identify the source of network congestion or other problems (e.g: a LAN or WAN issue)**
  - **Ability for application developers to characterize their own traffic, and how it is impacted by the network**
  - **Protocol analysis and debugging**
    - Often not possible to capture packet traces at the sending host

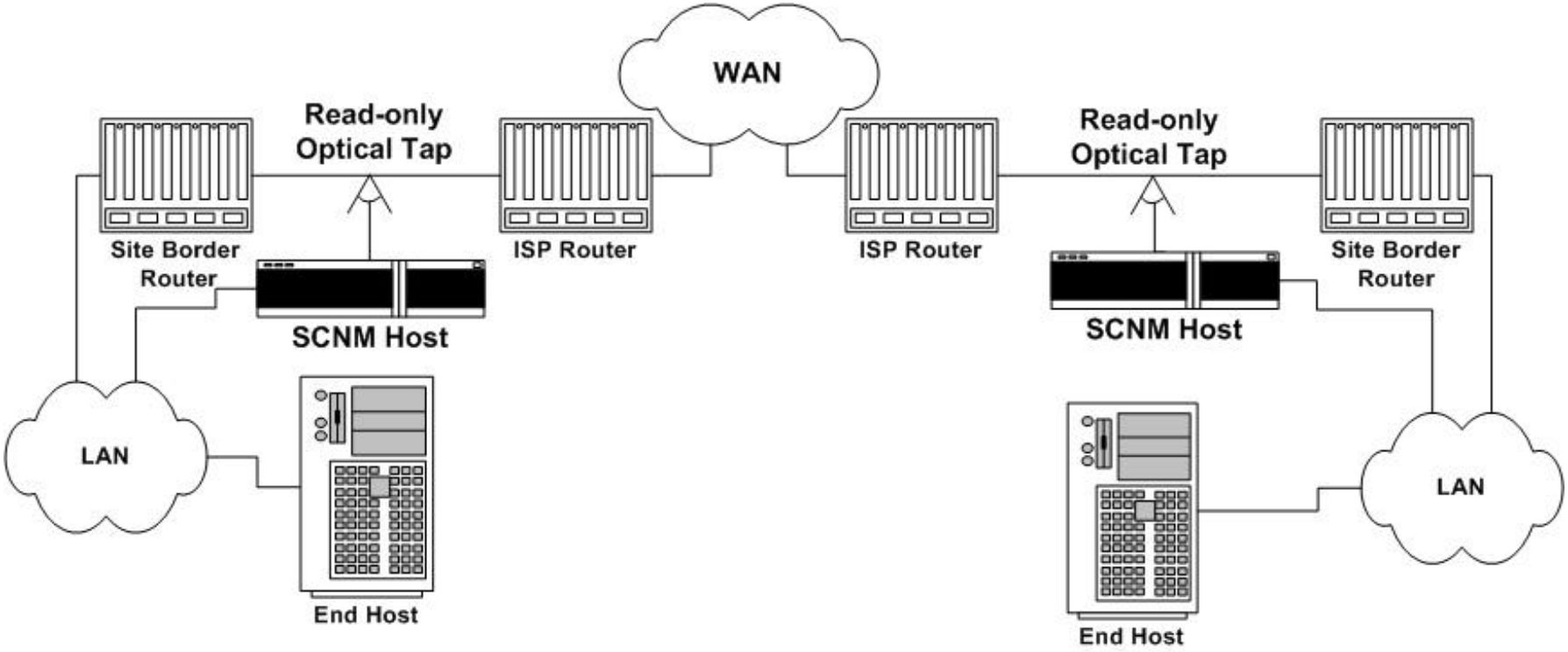
      ⇒tcpdump will often lose packets when trying to capture a high bandwidth stream

U.S. Department of Energy

Office of Science

Step 5: Add Passive Monitoring "Inside" the Network

BERKELEY LAB

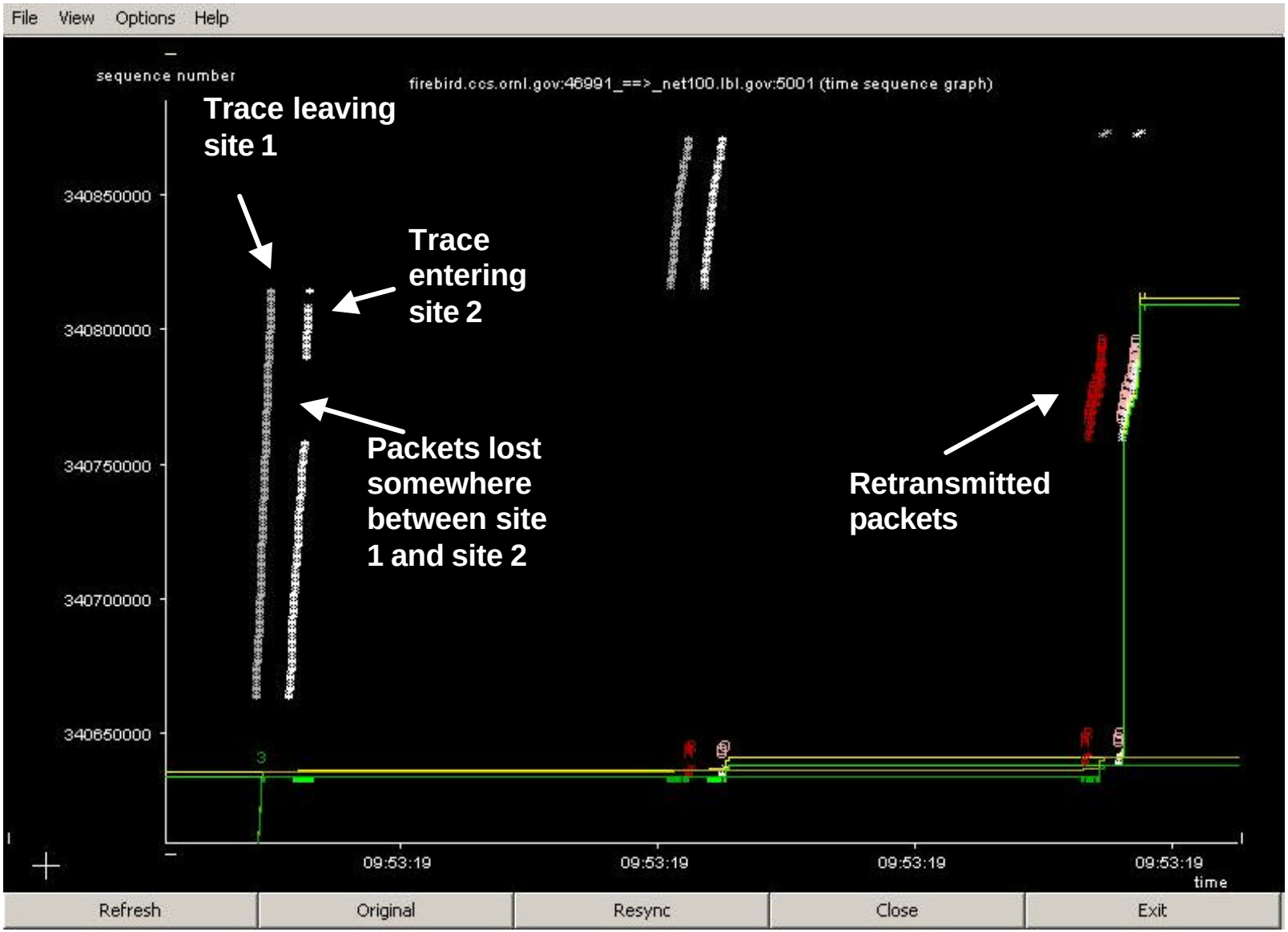


U.S. Department of Energy

Office of Science

Typical Passive Header Capture Results

BERKELEY LAB



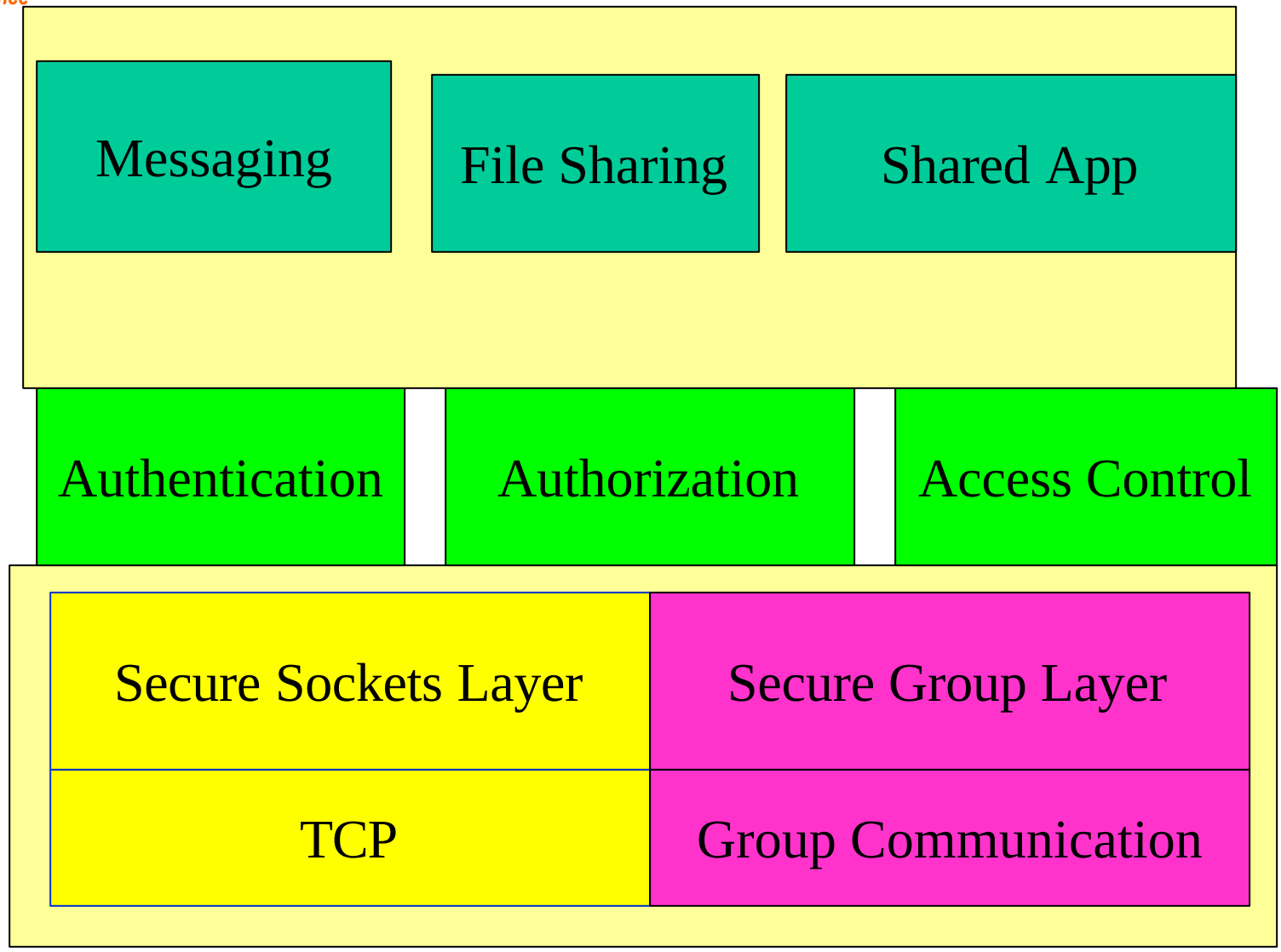
Conclusion

- Collaborative interactions need to be supported by a continuum of tools
- Required infrastructure/middleware is beginning to become available
- Scalability and security are important aspects that must be considered
- Monitoring is important for tracking what is happening
- Ease of use/adoption is critical

URL

- http://www-itg.lbl.gov/

Components

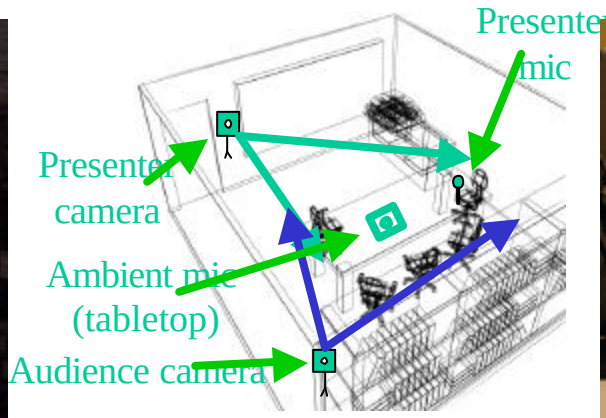

Messaging
File Sharing
Shared App
Authentication
Authorization
Access Control
Secure Sockets Layer
Secure Group Layer
TCP
Group Communication

U.S. Department of Energy

Office of Science

Access Grid Nodes (ANL)

BERKELEY LAB



Release Status

- Released code is available for
  - Akenti authorization library with C++ and C interfaces
  - Akenti standalone authorization server
  - support for runtime conditions
  - Java GUI and command line tools to create and verify policies
  - *http://www-itg.lbl.gov/Akenti/download.html*